Project: PID-Controller

Implementation

The implementation of the PID controller was more or less straightforward:

1. Initializations: All errors are set to zero. The parameters Kp, Ki and Kd are set to the values provided by the function input.
2. Update error: d_error is set to the difference of the current and previous cte values, p_error is set to the current cte value and i-error was set to the sum of all previous cte-values.
3. Total error: Total error is set to the negative sum of the p-error, d-error and i-error multiplied by the parameters Kp, Kd and Ki respectively.

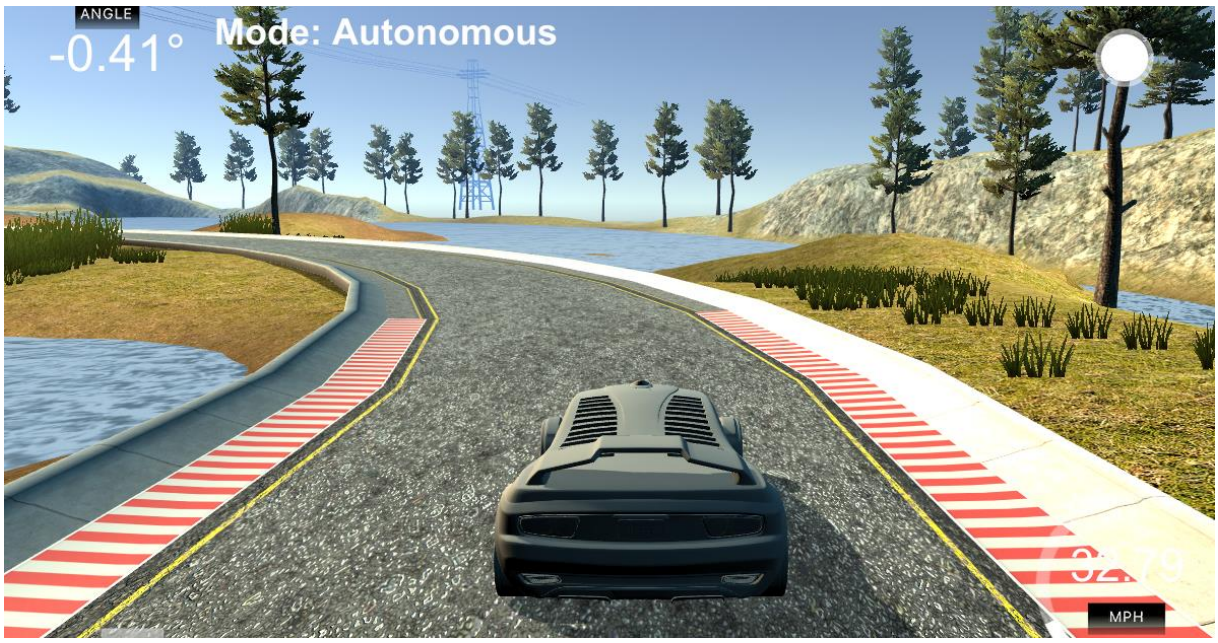
PID-Controller equation could be also implemented directly in the main file, since error variables and parameters are public. But I thought that it is better to access the class internal variables by the class own function like TotalError and not to access class variables directly from the main.cpp.

Parameter Optimization

Parameter optimization was most time consuming part. I chose an iteratively heuristic approach. At first I estimated effect of each controller type (P, I and D):

1. I activated only P-controller (Kp >0) and set all other parameters to zero. (Kd = Ki = 0). The car drove well on the strait sections but oscillation and overshoot led to drifting off in curves with P-control alone, see figure below:

2. Then I switched on D-component to get PD controller (Kp, Kd > 0, Ki = 0) and after some optimization of Kp and Kd parameters the car was able to drive well in not too sharp curves like in figure below.



But in sharp curves the car drifted off a little as can be seen in figure below.

3. Finally, I switched on I-component to get PDI-controller (Kp, Kd and Ki > 0) and after some optimization of parameters the car was able to drive well also in sharp curves like in figure below.

In the next steps I optimized parameters. The initial clue for the parameter setting come from Sebastian's videos where the parameter Kp is in order of magnitude 0.1, the parameter Kd is at least 10 times bigger and the parameter Ki is 100-1000 times lower. Then I changed parameter a little bit, check in video how it behaves and performed following iterative procedure for parameter optimization:

1. If the oscillations or overshooting was too high I increased Kd parameter.
2. If the car tended to drift from the road in sharp curves I increased the parameter Ki.
3. If the car didn't follow the curves I increased the parameter Kp.
4. If performance was ok leave the loop, otherwise go to step 1.

Conclusions

It was a nice experience for me to see the effect of each controller on car behavior visually in the simulator. Now I have much better feeling for controller functions than looking at the equations alone.